

Rechnersysteme 2

Praktikum 1

SoC

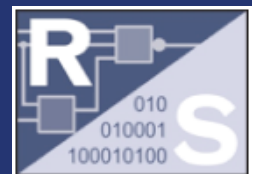
Seminararbeit vorgelegt von Marcel Mann und Marcel Humm

Abgabe: 15. Januar 2018

Institut für Datentechnik | Fachgebiet Rechnersysteme | Prof. Dr.-Ing. Christian Hochberger



TECHNISCHE
UNIVERSITÄT
DARMSTADT



1 Aufgabenstellung

Im Rahmen dieses Praktikums sollte ein auf Ultraschall basierender Abstandsmesser mit Display entstehen. Dazu wurde jeder teilnehmenden Gruppe ein System-on-Chip Kit (SoC Kit) zur Verfügung gestellt. Dessen Basis bildet das SP601, ein FPGA-Board der Firma Xilinx, erweitert mit einem SRF02-Ultraschallsensor und einem OLED Display Modul der Firma Newhaven. Zur Entwicklung des Abstandsmessers sollte zunächst das Software-Tool JConfig zur Konfiguration der gegebenen Hardware verwendet werden. Darauf aufbauend galt es die Software-Treiber für die I²C-Schnittstelle des Ultraschallsensors als auch die SPI-Schnittstelle des Displays zu implementieren. In einer Reihe von Zusatzaufgaben bestand weiterhin die Möglichkeit die notwendige Rechenleistung und Ausführungszeit der entwickelten Firmware durch die Verwendung von Interrupts und Code-Profilng zu optimieren.

2 Hardware

2.1 Ultraschallsensor

Beim SRF02 handelt es sich um einen kompakten Ultraschallsensor der Firma Devantech Ltd., der sich sowohl über einen I²C-Bus als auch über eine serielle RS232-Schnittstelle ansteuern lässt. Die verwendete Ultraschallfrequenz beträgt 40kHz. Der messbare Bereich des Sensors liegt abhängig von Umgebung und Temperatur zwischen ca. 15cm und 6m. Eine einzelne Messung kann dabei bis zu 65ms dauern. Da der Sensor während eines Messvorgangs keine weiteren Befehle annehmen kann, liegt dessen maximale Messfrequenz somit bei 15 Messungen pro Sekunde. Standardmäßig kann der Sensor über die Adresse 0xE0 angesteuert werden. Über ein beschreibbares Befehlsregister kann ein Messvorgang mit Auswertung der gemessenen Distanz in Zentimetern, Zoll oder Mikrosekunden (Zeit bis Echo) gestartet werden. Das Ergebnis der Messung und die aktuell messbare Minstdistanz können über vier weitere Register ausgelesen werden. Ein weiteres vorhandenes Register besitzt keine Funktion.

2.2 Display

Zur Ausgabe der vom Ultraschallsensor gemessenen Distanz das OLED-Display-Modul NHD-2.8-25664UCB2 mit einer Auflösung von 256 x 64 Bildpunkten der Firma Newhaven verwendet. Über den integrierten SSD1322-Controller von Solomon Systech kann das Display über eine serielle oder parallele Schnittstelle angesteuert werden. Für dieses Praktikum wird lediglich die serielle Schnittstelle (3-wire) mit einer Framebreite von 9 Bit verwendet.

2.3 FPGA-Board

Die Basis des verwendeten SoC-Kits bildet das SP601 Evaluation Kit der Firma Xilinx. Das um einen Spartan-6 FPGA aufgebaute Board bietet neben SPI- und I²C-Bussen auch Ethernet, USB zum Testen und Debuggen der Konfiguration mit JTAG und UART, diverse User-I/O-Pins, einen Oszillator zur Taktgenerierung, diverse Status-LEDs und 128 MB DDR2 RAM. Auf dem Spartan-6 FPGA wird der speziell für die Verwendung in FPGAs entwickelte 18 Bit SpartanMC-Softcore-Prozessor abgebildet.

3 Protokolle

3.1 I²C

Bei I²C, kurz für Inter-Integrated Circuit, handelt es sich um einen seriellen multi-Master und multi-Slave fähigen Datenbus. Zwei Signalleitungen ermöglichen die Kommunikation zwischen jeweils zwei Komponenten nach dem Master-Slave-Prinzip. Über die Serial-Data-Line (SDA) können Master und Slave Informationen austauschen. Der dazu benötigte Takt wird vom Master über die Serial-Clock-Line (SCL) gegeben.

Verbindungsaufbau

Bei Inaktivität des I²C-Bus liegt an beiden Signalleitungen eine 1 an. In diesem Zustand kann ein angeschlossener Master eine Verbindung zu einem angeschlossenen Slave herstellen, indem er SDA auf 0 setzt. Nach diesem Startsignal kann mit jeder steigenden Flanke des SCL ein Bit übertragen werden. Ein Datenpaket besteht dabei aus jeweils 8 Bit.

Datenübertragung

Während der Datenübertragung zwischen Master und Slave darf sich das an SDA angeschlossene Signal nur ändern, während an SCL eine 0 anliegt. Das zunächst gesendete Datenpaket sollte die 7 Bit lange Adresse des zu adressierenden Slaves enthalten. Das achte Bit dieses Startpakets signalisiert die Lese- (1) oder Schreibabsicht (0) des Masters. Der Erhalt eines Datenpakets muss vom Empfänger mit einem Acknowledge-Bit bestätigt werden. Eine 0 signalisiert hierbei eine erfolgreiche Übertragung.

Verbindungsabbruch

Nach Abschluss der Datenübertragung kann der I²C-Bus wieder in einen inaktiven Zustand gesetzt werden, indem SDA von 0 auf 1 geändert wird, während an SCL eine 1 anliegt.

3.2 SPI

Das Serial Peripheral Interface (SPI) beschreibt eine serielle voll duplex-fähige Kommunikationsschnittstelle über die sich ein einzelner Master mit einer Menge unterschiedlicher Slaves verbinden lässt. Alle angeschlossenen Komponenten teilen sich die drei Signalleitungen serial clock (SCLK), master out slave in (MOSI) und master in slave out (MISO). Zusätzlich existiert für jeden angeschlossenen Slave eine fest zugeordnete low-active slave select (SS) Leitung.

Eine Datenverbindung beginnt, indem der Master das SS-Signal des zu adressierenden Slaves auf 0 setzt und anschließend mit der Generierung eines Takt-Signals beginnt. Analog zu I²C wird auch hier mit jedem Takt ein Bit übertragen. Da jedoch bei SPI nur ein einzelner Master existiert, können Wortlänge, Taktfrequenz und Phasenlage (Datenübertragung bei steigender gegenüber fallender Taktflanke) von diesem an die jeweiligen Bedürfnisse unterschiedlicher Slaves angepasst werden. Den Wortanfang kann der ausgewählte Slave durch den vom Master gegebenen Takt erkennen.

4 Vorgehen

- 1) Ultraschallsensor aktivieren (induziert durch aufleuchten der LED)
- 2) Messwerte in cm auslesen und auf Konsole ausgeben
- 3) Filtern ungenauer Messwerte
- 4) Ein- und Ausschalten des Displays
- 5) Ausgabe einfacher Strings auf dem Display
- 6) Ausgabe der ausgelesenen Messwerte auf dem Display

Um eine strukturierte Durchführung dieses Praktikums zu gewährleisten, haben wir uns zunächst entschieden die gegebene Aufgabenstellung in 6 aufeinander aufbauende Meilensteine zu gliedern.

1) Aktivieren des Ultraschallsensors

Da der Ultraschallsensor gemäß Aufgabenstellung über dessen I²C-Schnittstelle angesteuert werden sollte, galt es zunächst das I²C-Master-Modul des SpartanMC mit Hilfe von JConfig zu konfigurieren. Im nächsten Schritt wurde ein einfacher Treiber implementiert, der dem I²C-Master-Modul eine Reihe von Anweisungen übergibt. Um den Ultraschallsensor zu aktivieren, muss das I²C-Master-Modul zunächst gestartet und dabei ein zu verwendender Takt gegeben werden. Anschließend gilt es den Ultraschallsensor als Slave zu adressieren und in dessen Befehlsregister den Start-Befehl zur Entfernungsmessung in Zentimetern zu setzen. Die erfolgreiche Messung wird am Ultraschallsensor durch das Aufleuchten einer LED signalisiert.

2) Messwerte auslesen

Den begonnenen Treiber galt es im nächsten Schritt zu erweitern, um eine Einsicht der gemessenen Distanz des Ultraschallsensors zu ermöglichen. Da dieser gemäß seiner Spezifikation bis zu 65ms benötigt bis die Messwerte nach Beginn der Messung in den jeweiligen Registern abgelegt wurden, muss zunächst über einen sleep-Befehl entsprechend lange mit dem Auslesen gewartet werden. Anschließend gilt es den Ultraschallsensor erneut als Slave zu adressieren und diesem die mitzuteilen, welche Register der Master im nächsten Schritt auslesen möchte. Darauf folgt schließlich die Adressierung des Ultraschallsensors als Slave mit Lesebefehl. Nach der Übertragung der angegebenen Registerinhalte liegen diese in den Datenregistern des I²C-Master-Moduls ab und können vom Treiber über die UART-Schnittstelle auf der Konsole ausgegeben werden

5 Implementierung

Die Erläuterungen zu unserer Implementierung, teilen wir in verschiedenen Abschnitten ein.

5.1 Ultraschallsensor / I²C

Der erste Schritt bestand darin den I²C-Master einzurichten und darüber den Ultraschallsensor auszulesen. Der I²C-Master wird über seine externen Register angesprochen. Zuerst wird er über das Setzen des Enable-Bits und das Festlegen des Prescalers aktiviert. Der Prescaler wird nach der Formel $Prescaler = peripheral_clock / (5 * desired_scl) - 1 = 60MHz / (5 * 40KHz) - 1 = 299$ bestimmt. Ist der I²C-Master aktiviert, sind drei Schritte notwendig um einen Abstandswert zu erhalten:

- 1. Anweisung zum Starten der Messung senden
- 2. Zu lesendes Register an den Sensor senden
- 3. Die beiden Ergebnisregister auslesen

Da der Abstandswert auf 2 Register aufgeteilt ist, muss der finale Abstandswert noch errechnet werden.

5.2 OLED-Display / SPI

Auch der SPI-Master muss zuerst über seine Register aktiviert werden. Um das Display zu aktivieren, muss ihm eine Reset-Sequenz gesendet werden. Ist dies geschehen, ist es möglich dem Display über die SPI-Schnittstelle Daten zu senden.

5.3 Software-Design

Unsere Software haben wir nach ihren Aufgabengebieten unterteilt. Zur Ansteuerung der Schnittstellen haben wir jeweils eine Treiber-Datei implementiert, samt dazugehörigen Headern. Weitestgehend unverändert haben wir den Programmcode zur Steuerung des Displays übernommen, da dieser schon viele nützliche Funktionen bereitstellt. Zusätzlich haben wir eine Quellcodedatei mit Hilfsfunktionen und Datenstrukturen angelegt.

5.4 Ringbuffer

Um Messwerte vor der Ausgabe zwischenspeichern zu können, haben wir einen Ringbuffer implementiert. Ein Ringbuffer besteht aus einem Datenarray, 2 Pointern head und tail, welche die Anfangs- und endpositionen der Daten im Array markieren und einem counter, welche die Anzahl der aktuell im Array gespeicherten Werte festhält. Um den Buffer einfacher nutzen zu können, haben wir die Funktionen *put*, *get*, *isFull* und *isEmpty* implementiert.

5.5 Messwertfilter

Die vom Ultraschallsensor ausgelesenen Messwerte können durch verschiedene Faktoren negativ beeinflusst werden. Neben statischen Fehlern wie der Verwendung einer falschen Berechnungsformel im Treiber zählen dazu auch dynamische Fehlerquellen basierend auf den physikalischen Eigenschaften der Ultraschallmessung. Um fehlerhafte Ausgaben zu vermeiden, wurde ein einfacher Ringspeicher implementiert. In diesem werden fortlaufend die letzten X Messergebnisse zwischengespeichert. Mit jedem eingehenden Ergebnis wird zunächst das älteste gespeicherte Ergebnis aus dem Ringspeicher verdrängt. Anschließend wird der aktuelle Inhalt des Ringspeichers kopiert, sortiert und dessen Median als aktuelle Distanz ausgegeben.

Weiterhin werden nur Werte in den Ringspeicher übernommen, die zwischen der kleinsten und größten messbaren Distanz des Ultraschallsensors liegen. Während es sich bei der größten messbaren Distanz um einen statischen Wert von 594cm handelt, ist die kleinste messbare Distanz abhängig von Umgebungsvariablen wie beispielsweise der Raumtemperatur. Beim Start der Programmausführung wird die aktuell kleinste messbare Distanz dementsprechend über die I²C-Schnittstelle vom Ultraschallsensor abgefragt und als Untergrenze für den Eintritt von Messwerten in den Ringspeicher übernommen.

5.6 Ausgabe

Um den Wert unter Nutzung der gegebenen Methode auf dem Display auszugeben, muss der erhaltene Integer-Wert in eine nulterminierte Zeichenkette umgewandelt werden. Diese Umwandlung geschah unter Nutzung der *snprintf*-Methode. Um nicht nur einen Zahlenwert auszugeben, soll der Messwert in einen String der Art "Distance: XXXcm" umgewandelt werden. Eine dazu nötige *append*-Methode wurde selbst implementiert. Der fertige String kann nun an die OLED-Ausgabemethode und an *printf* zur UART-Ausgabe via USB übergeben werden.